



**¡Les damos la  
bienvenida!**

¿Comenzamos?

Esta clase va a ser

- grabada



COMISIÓN N°####

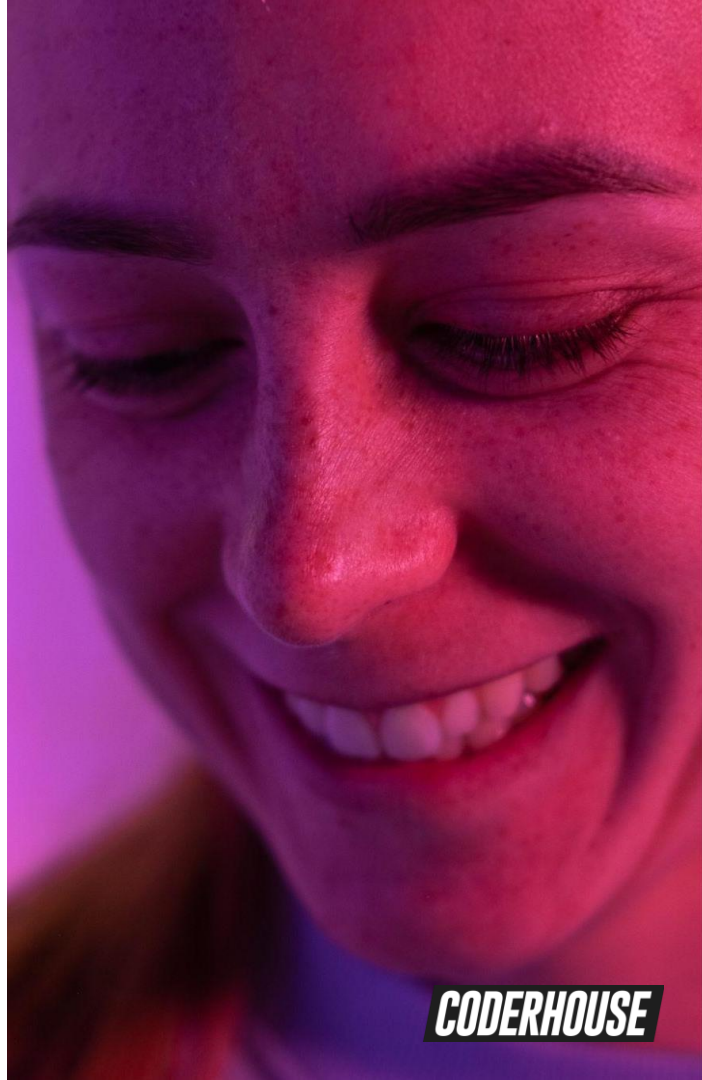
# Presentación del equipo

- ✓ Profesor: Emanuel Tevez
- ✓ Tutores: Nicolas Dziuma

# Presentación de estudiantes

Por encuestas de Zoom

1. País
2. Conocimientos previos
3. ¿Por qué elegiste este curso?



**CODERHOUSE**



Lo que debes saber  
**antes de empezar**

# Acuerdos y compromisos

## ACUERDOS Y COMPROMISOS

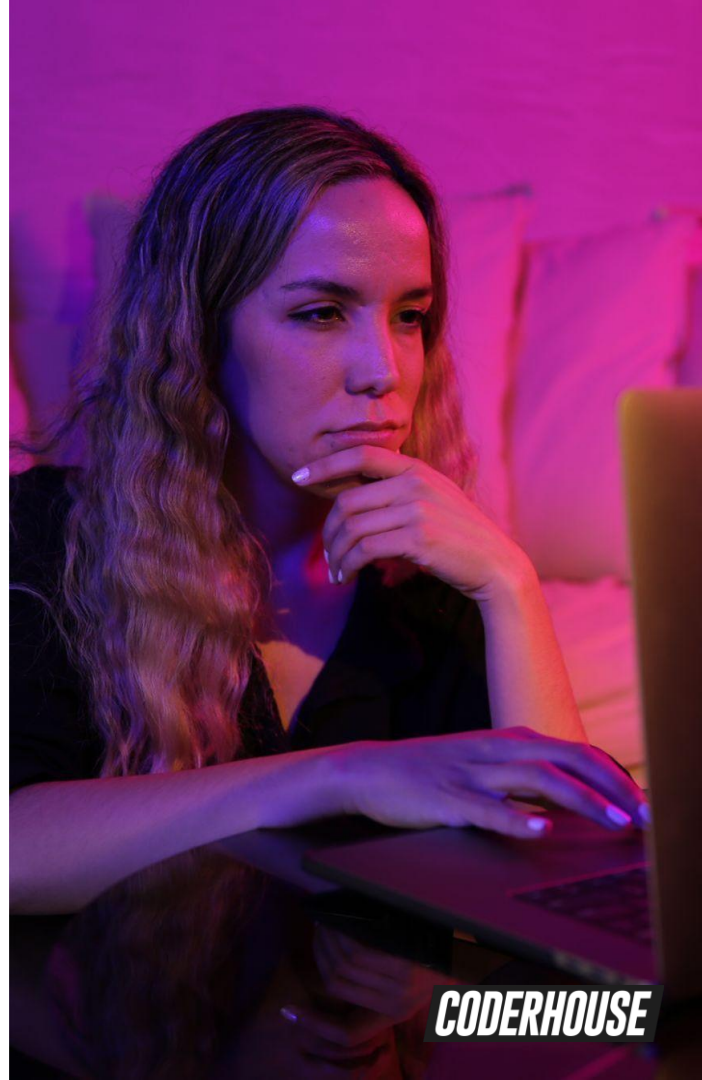
# Convivencia

- ✓ Conoce aquí nuestro [código de conducta](#) y ayúdanos a generar un ambiente de clases súper ameno.
- ✓ Durante las clases, emplea los medios de comunicación oficiales para canalizar tus dudas, consultas y/o comentarios: **chat Zoom público y privado, y por el chat de la plataforma.**
- ✓ Ten en cuenta [las normas del buen hablante y del buen oyente](#), que nunca están de más.
- ✓ Verifica el estado de **la cámara y/o el micrófono** (on/off) de manera que esto no afecte la dinámica de la clase.

ACUERDOS Y COMPROMISOS

# Distractores

- ✓ Encuentra tu espacio y crea el momento oportuno para **disfrutar de aprender**
- ✓ Evita dispositivos y aplicaciones que puedan **robar tu atención**
- ✓ Mantén la **mente abierta y flexible**, los prejuicios y paradigmas no están invitados



**CODERHOUSE**



## ACUERDOS Y COMPROMISOS

# Herramientas

- ✓ Mantén a tu alcance **agua, mate o café**
- ✓ Si lo necesitas, ten a mano lápiz y papel para que no se escapen las ideas. Pero recuerda que **en Google Drive tienes archivos que te ayudarán a repasar, incluidas las presentaciones.**
- ✓ Conéctate desde algún equipo (laptop, tablet) que te permita **realizar las actividades** sin complicaciones.
- ✓ Todas las clases quedarán grabadas y serán compartidas tanto en la **plataforma de Coderhouse como por Google Drive.**



## ACUERDOS Y COMPROMISOS

# Equipo

- ✓ Intercambia ideas por **el chat de la plataforma**.
- ✓ Siempre **interactúa respetuosamente**.
- ✓ No te olvides de **valorar tu experiencia** educativa y de contarnos cómo te va.

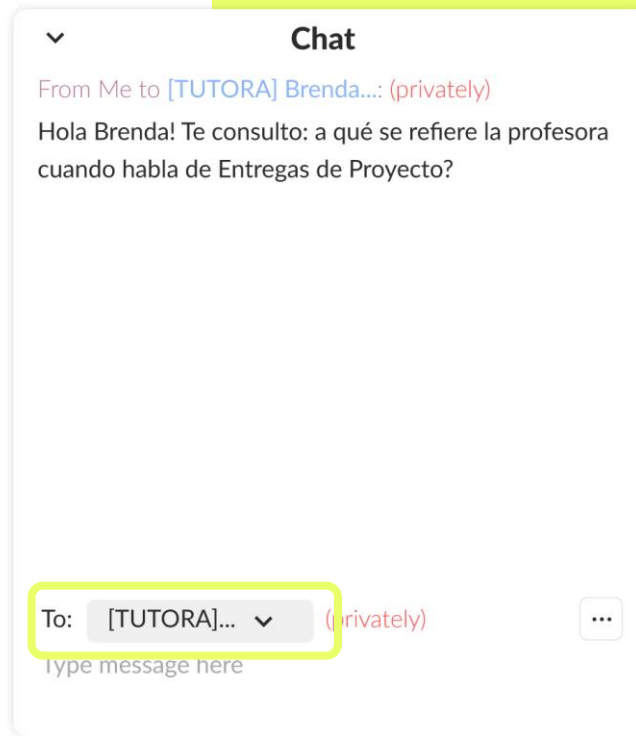
# Interacciones en clase

## INTERACCIONES EN CLASE

# Mientras el profesor explica

Para mantener una comunicación clara y fluida a lo largo de la clase, te proponemos mantener 2 reglas:

1. Si tienes dudas durante la explicación, debes consultarle directamente por privado a tu tutor por el chat de Zoom.

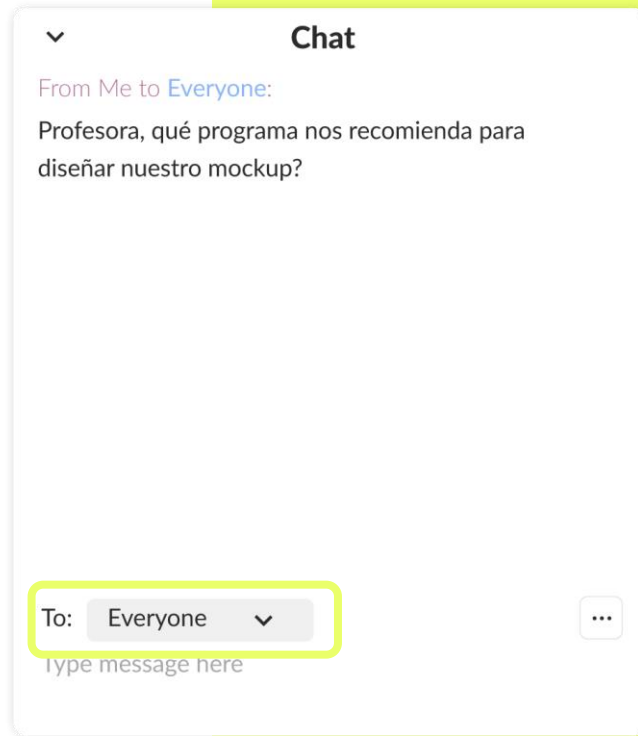


# Espacios para consultas

2. Entre contenido y contenido, se abrirán breves espacios de consulta. Allí puedes escribir en el chat tu pregunta.

**¡Tu duda puede ayudar a otras personas!**

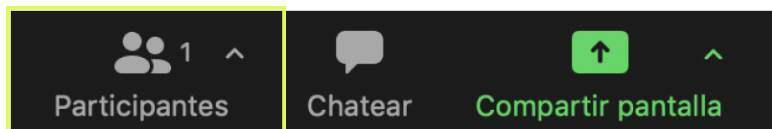
No olvides seleccionar “todos” para que todos puedan leerte (y no solo tu tutor).



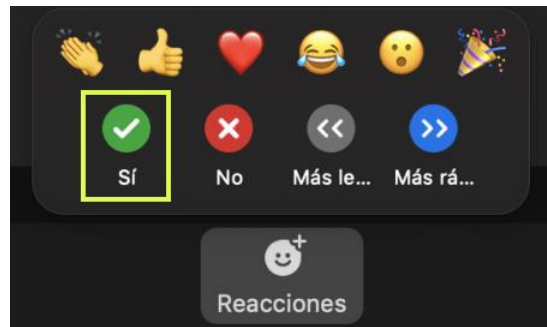
## INTERACCIONES EN CLASE

# Funcionalidades

Para **evitar saturar el chat de mensajes**, utiliza los signos que figuran en el apartado **Participantes**, dentro de Zoom.\*\*



**Por ejemplo:** si se pregunta si se escucha correctamente, debes seleccionar la opción "Sí" o "No".



\*\*Para quitar el signo, presiona el mismo botón nuevamente o la opción "clear all".



¿Cuál es nuestro  
Proyecto final?



# ¿Qué es una Web Playground?

Un playground es un **lugar de pruebas**, donde podremos hacer, deshacer, refactorizar, romper, etc.

En un Playground **no se abordan las cuestiones relativas a la Interfaz Gráfica (la parte visual de la App), solo trabajaremos con el Código Fuente**. La idea de generar una web playground es para que podamos aventurarnos en las funciones más avanzadas de Python y Django, de tal forma que siempre tengamos un lugar donde probar dichas funciones.





## PROYECTO FINAL

# Web Playground

### Consigna:

Desarrollarás una **web** en Django. Las entregas son individuales. Se verán temas avanzados:

### Registros:

- ✓ Login/signup
- ✓ Reset pwd
- ✓ Logout

### Perfil:

- ✓ Imagen
- ✓ Editar email/pwd

### Páginas:

- ✓ CRUD solo si está registrado

### Admin:

- ✓ Apps





PROYECTO FINAL

# Proyectos de nuestros estudiantes

También puedes ver el Proyecto Final de [Johannes Pérez](#) y [Germán López](#), estudiantes de este curso de comisiones anteriores.

- [Playground Final Project | Johannes Pérez](#)
- [Playground Final Project | Germán López](#)

¡Esperamos que te resulten inspiradores!



Clase práctica en vivo – Unidad 1

# Introducción a la programación con Python

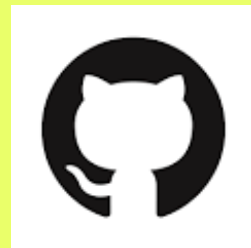
# Objetivos de la clase

- Definir Python como lenguaje de programación con sus principales ventajas y desventajas
- Presentar e instalar el stack de herramientas de trabajo
- Comenzar a implementar el entorno de programación "Colabs"
- Reconocer funcionalidades de cadenas de caracteres
- Implementar operaciones básicas y avanzadas de números

# Repositorio Github

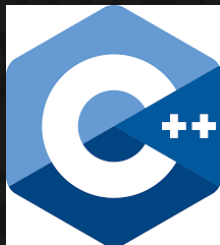
Te dejamos el acceso al Repositorio de Github donde encontrarás todo el material complementario y scripts de la clase.

✓ [Repositorio Python](#)



# Programación: un breve viaje por la historia

# ¿Cuáles son los más conocidos?





## Para pensar

¿Qué lenguajes de programación reconoces tras ver los logotipos?

Contesta en el chat de Zoom



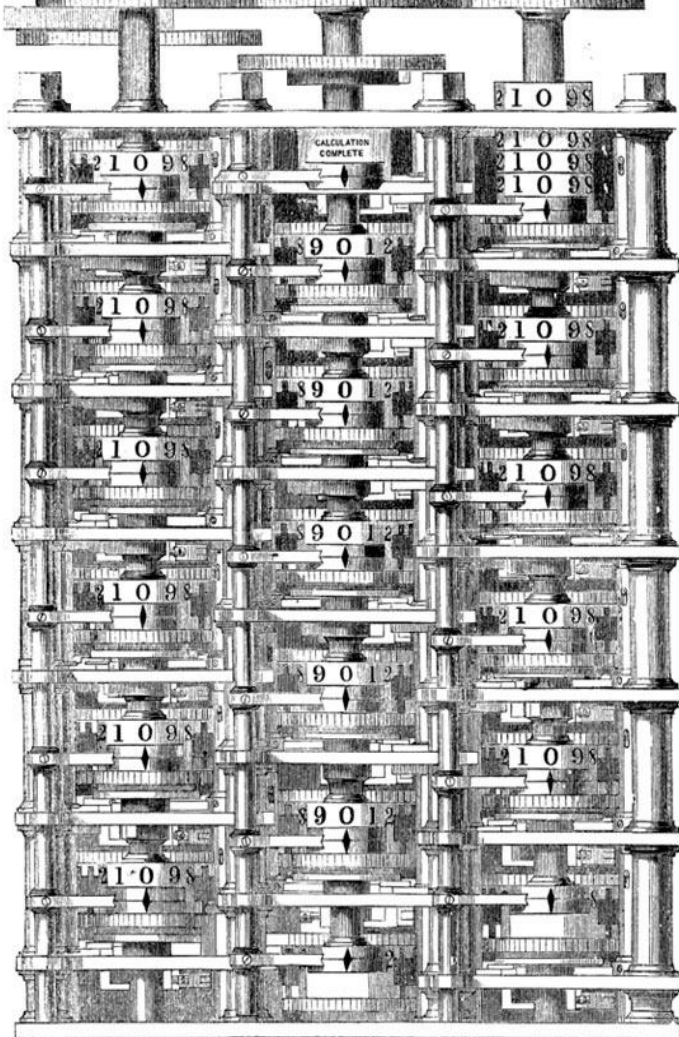


Programación  
en

*Ada*

# ¿Sabías qué...?

Lady Ada Countess of Lovelace (1815–1852), matemática y escritora, es reconocida como la **primera programadora de la historia**.



PORTION OF BABBAGE'S DIFFERENCE ENGINE.

# ¿Sabías qué...?

En su honor, Jean Ichbiah, en 1979, le puso **ADA** al lenguaje que había creado.



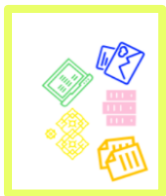
## Para pensar

Dato vs. Información

¿Datos e información son sinónimos? ¿Por qué?

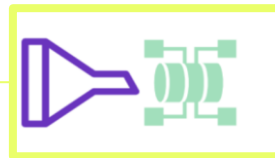
Contesta mediante el chat de Zoom

# Datos vs. Información



## Datos

Característica o atributo sin procesamiento, el cual no informa nada por sí solo.



## Información

Unión de datos procesados, que se complementan para informar un hecho.



# Break

¡10 minutos y volvemos!

# ¿Pudiste instalar Visual Studio Code?

- ¿Crear una carpeta?
- ¿Crear un archivo .py?
- ¿Ejecutar un archivo Python?

Python



# ¿Qué es?

Es un lenguaje de programación de código abierto, orientado a objetos, simple y fácil de entender.

Tiene una sintaxis sencilla que cuenta con una vasta biblioteca de herramientas, que hacen de Python un lenguaje de programación único.

Es un **lenguaje interpretado con tipado dinámico, multiparadigma y multiplataforma.**



# Google Colab

Para programar con comodidad se suele instalar un IDE o editor de texto. En este curso les proponemos instalar [Visual Studio Code](#); pero lo vamos a usar a partir de la clase 17.

👉 Por ahora trabajaremos con un entorno de **Google** para enfocarnos solo en la programación, este entorno se llama **Colab**.





# ¿Cuánto aprendiste?

Realizaremos un repaso por los temas más importantes de la clase

Duración: **10 minutos**



## ACTIVIDAD EN CLASE

# ¿Cuánto aprendiste?

### Descripción de la actividad.

De manera individual, realizaremos un repaso por los temas más importantes de la clase de hoy.

Para ello, deberás ingresar a [este link](#).

Luego, comparte tus aciertos por el chat 😊

Recuerda que podrás volver a ver esta clase para reforzar los temas que necesites.



# Tu primer programa...

en Visual Studio Code.

Duración: **10 minutos**



## ACTIVIDAD EN CLASE

# Mi primer programa en Visual Studio Code

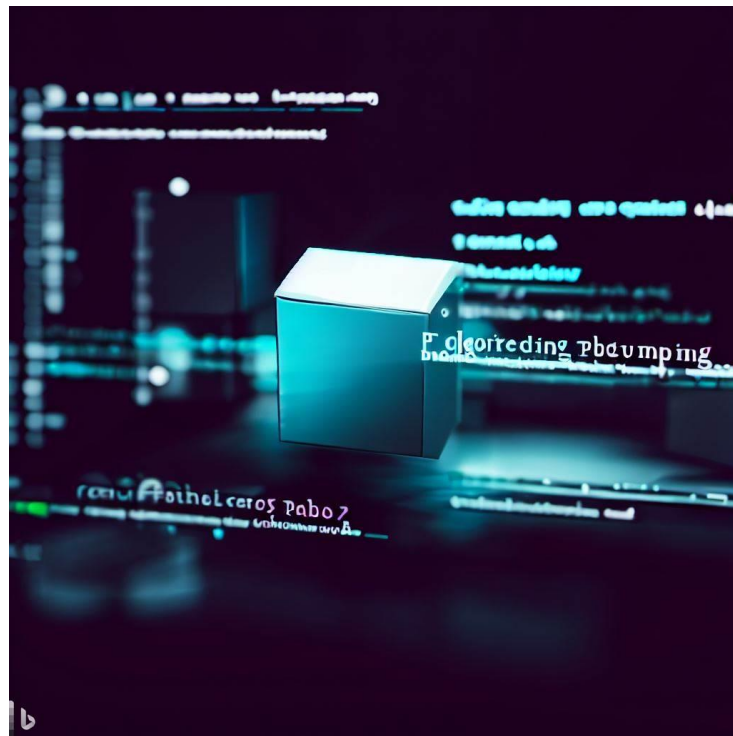
1. Crear una carpeta llamada Coderhouse y abrirla con Visual Studio Code.
2. Crear un nuevo archivo, con el nombre: `clase_01.py`
3. Ahora, en el código: mostrar por pantalla un mensaje de bienvenida
4. Preguntar al usuario por su nombre y luego por su edad, guardando los valores en dos variables.
5. Mostrar por pantalla en un solo mensaje: "nombre" tiene "edad" años.  
Siendo nombre y edad las variables que ya tienen un valor.

# Variables

# Variables

En algunos lenguajes de programación, las variables se pueden entender como "cajas" en las que se guardan los datos.

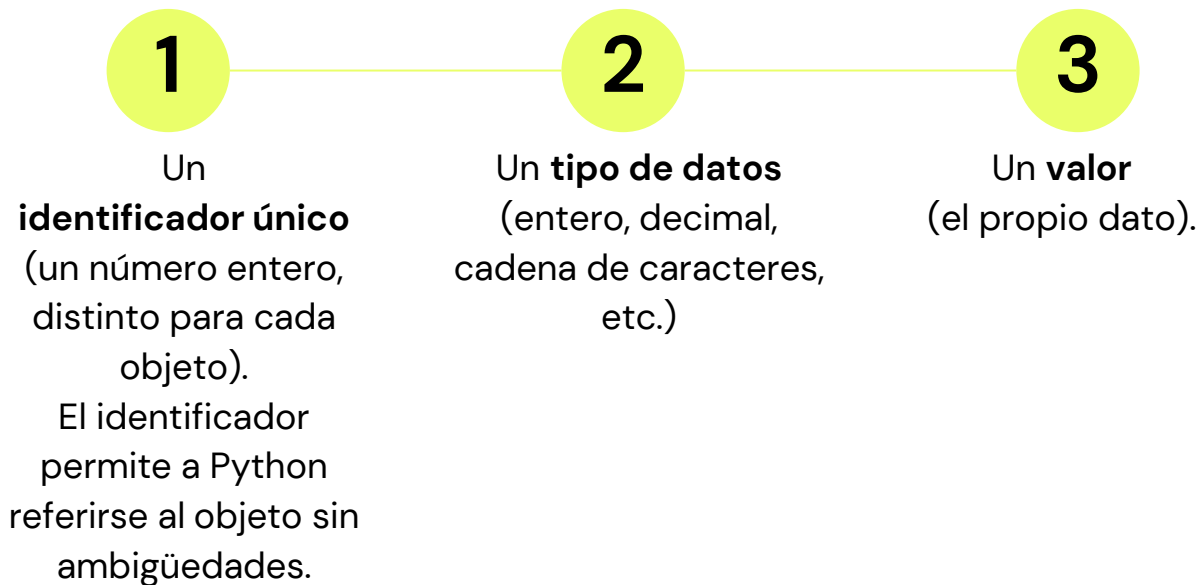
Pero en Python las variables son **"etiquetas"** que **hacen referencia a los datos** (que se guardan en unas "cajas" llamadas **objetos**).



# Variables en programación

Por cada dato que se crea en un programa, Python crea un **objeto** que lo contiene.

Cada objeto tiene:







PARA RECORDAR

# Variables en Python

Las variables en Python no guardan los datos, sino que son simples nombres para **poder hacer referencia a esos objetos.**

# Variables en programación

En Python, si escribimos la instrucción:

```
a = 2
```

Se crea el objeto 2.

Ese objeto tendrá un identificador único que se asigna en el momento de la creación y se conserva a lo largo del programa.

Se asocia el nombre "a" al objeto número entero 2.





# Definir una variable

Siempre se escribe a la izquierda de la igualdad, de lo contrario, Python generará un mensaje de error:

❌ `2 = mi_variable`

Para mostrar el valor de la variable hay que imprimir su nombre.

✨ `mi_variable = 2`  
`print(mi_variable)`



# Definir una variable

Si una variable no se ha definido previamente, al escribir su nombre o imprimir la variable generará un error:

```
x = -10
```

```
❌ y
```



# Definir una variable

Una variable puede almacenar números, texto o estructuras más complicadas (que se verán más adelante). Si se va a almacenar texto, debe escribirse entre comillas simples (') o dobles ("), que son equivalentes. A las variables que almacenan texto se les suele llamar cadenas (de texto).

```
mi_auto = "ford"  
print(mi_auto)
```



# Definir una variable

Si no se escriben comillas, Python supone que estamos haciendo referencia a otra variable (que, si no está definida, genera un mensaje de error):

❌ `mi_auto = ford`

OK `ford = "Ford Mustang"`

OK `mi_auto = ford`

OK `print(mi_auto)`



# Veamos un ejemplo en vivo

El nombre de una variable debe empezar por una letra o por un guion bajo (\_) y puede seguir con más letras, números o guiones bajos. Los nombres de variables no pueden incluir espacios en blanco.

Se recomienda que sea todo con minúsculas, separadas las palabras con guiones bajos. Este estilo se llama **snake\_case**.



```
FecNac = "27 de octubre de 1997"
```



```
fecha_de_nacimiento = "27 de octubre de 1997"
```



# Veamos un ejemplo en vivo

Los nombres de las variables pueden contener mayúsculas, pero ten en cuenta que **Python distingue entre mayúsculas y minúsculas** (en inglés se dice que Python es case-sensitive).

```
nombre = "Pepito Conejo"  
Nombre = "Numa Nigerio"  
nomBre = "Fulanito Mengáñez"  
nombre
```





# Desafío String

Genera una nueva variable

Duración: 7 minutos



ACTIVIDAD EN CLASE

# Desafío String

Partiendo de:

```
cadena_1 = "versátil"  
cadena_2 = "Python"  
cadena_3 = "es un lenguaje"  
cadena_4 = "de programación"
```

Crea una nueva cadena con el siguiente contenido:

**"Python es un lenguaje de programación versátil"**

Nota: Utiliza el operador + para unir las cadenas (concatenación)



**CODERHOUSE**



# Desafío números

De manera individual desarrollarán un programa que permita calcular el promedio final de puntos de los equipos de fútbol en un torneo.

Duración: **12 minutos**



## ACTIVIDAD EN CLASE

# Desafío números

### Descripción de la actividad.

En nuestro trabajo, nos solicitan desarrollar un programa que permita calcular el promedio final de los equipos de futbol en un torneo. Para ello, debemos considerar tres aspectos:

- ✓ Jugaron 20 partidos durante el torneo.
- ✓ Los partidos poseen diferente puntaje según el resultado, los partidos ganados 3 puntos, partido empatado 1 punto, partido perdido 0 puntos.
- ✓ El promedio final resulta de la cantidad de puntos totales divididos la cantidad de partidos



## ACTIVIDAD EN CLASE

# Desafío números

### Descripción de la actividad.

La cantidad de partidos que debemos considerar a un equipo para el ejemplo se detallan a continuación:

```
partidos_ganados 8  
partido_empatados 7  
partido_perdidos 5
```

**Importante:** Cada una de las cantidades de partidos ganados, empatados o perdidos debe solicitarse al usuario utilizando la función **input()**.

Actividad extra



# Desafío Slicing

Se tiene una cadena de texto, pero al revés. Al parecer contiene el nombre de un alumno, la nota de un examen y la materia.

```
cadena = "acitametaM ,5.8 ,otipeP ordeP"
```

1. Dar vuelta la cadena y asignarla a una variable llamada **cadena\_volteada**. Para devolver una cadena dada vuelta se usa el tercer índice negativo con slicing: **cadena[::-1]**
2. Extraer nombre y apellido, almacenarlo en una variable llamada **nombre\_alumno**



## Actividad extra



3. Extraer la nota y almacenarla en una variable llamada nota.
4. Extraer la materia y almacenarla en una variable llamada materia.
5. Mostrar por pantalla la siguiente estructura, usando la concantenación de cadenas:

**NOMBRE APELLIDO ha sacado un NOTA en  
MATERIA**



# Resumen de la clase hoy

- ✓ Programación
- ✓ Python
- ✓ Stack inicial del programador: Python, Colab.
- ✓ Metodologías Ágiles, XP
- ✓ Números, Strings, Print
- ✓ Variables
- ✓ Index & Slicing



**Opina y valora**  
**esta clase**

¿Preguntas?

**Muchas gracias.**

**#EducaciónDigitalParaElMundoReal**